View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Louisiana State University

Louisiana State University
LSU Digital Commons

LSU Master's Theses | Graduate School

2011

Fault detection in process control plants using principal component analysis

Vamshi Krishna Kandula
Louisiana State University and Agricultural and Mechanical College, vkandu1@lsu.edu

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_theses

 Part of the Electrical and Computer Engineering Commons

Recommended Citation
Kandula, Vamshi Krishna, "Fault detection in process control plants using principal component analysis" (2011). *LSU Master's Theses*. 1800.
https://digitalcommons.lsu.edu/gradschool_theses/1800

This Thesis is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Master's Theses by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.

FAULT DETECTION IN PROCESS CONTROL PLANTS USING PRINCIPAL COMPONENT ANALYSIS

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

in

The Department of Electrical Engineering

by
Vamshi Krishna Kandula
Bachelor of Technology (Electrical and Electronics)
VIT University, 2009
December 2011

Dedicated to my parents Shakuntala Devi and Pattabhi Rama rao

ACKNOWLEDGEMENTS

First of all, I would like to thank Dr. Kemin Zhou for his help and guidance throughout the project. I feel that I have learned a great deal about with his assistance. I would like to thank both Dr. Guoxiang Gu and Dr. Xue-Bin Liang for serving as members of my examining committee.

Second, I would like to thank Dr.F.Carl Knopf for financial and moral support throughout my studies at LSU.

Finally, I would like to thank everyone from the LSU Department of Electrical and Computer Engineering and Department of Chemical Engineering. During my stay here as a graduate student, I feel that I have gained a great deal of knowledge, and I have enjoyed my time here.

TABLE OF CONTENTS

NOMENCLATURE

AEM:	Abnormal Event Management
MSM:	Multivariate Statistical Methods
PC:	Principal Components
PCA:	Principal Component Analysis
PCFC:	Process Control Flow Charts

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

ABSTRACT

The aim of this thesis is to use a statistical method (principal component analysis) to detect a fault in a system. PCA is a dimensionality reduction technique that is used here. First the PCA algorithm was implemented on a simple two dimensional random data to reduce its dimensionality. Then the algorithm was implemented on a three dimensional data from a linear first order process and introduce some white noise later on and compare the results with that under normal operating conditions, thus determining the presence of a fault. Then data from a real time plant is considered for testing. This is done by considering a Simulink model of the Tennessee Eastman challenge problem from downs and vogel's, which has around 42 output variables. The crucial step in a dimensionality reduction technique is determining the number of principal components. When we have data in two dimensions or three dimensions it is easy to figure out the number of principal components by simply looking at the eigenvalues of the covariance matrix. However when there is a huge data it becomes difficult to find the number of principal components by inspection. Here, the number of principal components is found by using parallel analysis. The simulation results are carried out with a one fault at a time and the results are compared with the plots under normal operating conditions. The delay times of some of the faults is tabulated (The delay time to detect the fault largely depend on the number of principle components, so with a different approach to find the principal components the delay times may vary from what we got here). Finally, a case study of one of the faults (IDV1) is done.

CHAPTER 1

INTRODUCTION

1.1 BACK GROUND AND LITERATURE REVIEW

Fault detection has been studied since a longtime. There are many scholarly articles on fault detection and diagnosis. Its importance lies in the fact that detecting a fault in time while the system is still running avoids abnormal events and losses, thus avoiding major system breakdowns and catastrophes. There are several ways that the faults can be detected. It can be divided into two major classes, *model based* and *data driven*. In model based fault detections one has to know the exact process model or the mathematical model of the system. Sometimes it is very difficult to know the exact process model of the system, in such cases data driven techniques can be used for fault detection and diagnosis. Many process control plants have large data archives of the plant parameters like pressure, temperature, flow-rate, etc. during normal and faulty operating conditions. This large data can be used to build a statistical model which can be used to detect the faults in the future. The larger the data the more accurate will be the results given by the statistical model that is built.

In this report, I tried to use the statistical method, Principle Component Analysis (PCA) to build a model to detect the faults. This can be used in places where we have a plant with *almost* constant process and have a huge data during normal operation and faulty operations of the plant. This is usually seen in the process control plants which are common in the real control

systems. The controller's main task is to guarantee the stability of the plant. The influence of disturbance, noise and perturbation has to be minimized and also to get an optimized control performance. There are many processes in real world, like reactors, heat exchanger pumps and compressors. The variables to be controlled here are mostly temperature, pressure, reactor speed and water level. With all these processes and process variables the data from a process control plant will be huge and also detecting the faults in the right time will be challenging.

1.1.1 Definition of Fault

Fault is an unaccepted deflection of at least one parameter or property of the system from the normal or standard conditions. When a variable that is being measured exceeds the threshold values then it said that there is a FAULT in the system. The threshold values can be known from the previous data taken during normal operation conditions.

1.1.2 Importance of Fault Detection

Fault detection is of great importance mainly where the operations are done remotely or in hazardous environment. Detecting the faults in time saves lot of time and money in repairing the equipment or the manufactured product.

1.1.3 Multivariate Statistical Methods (MSM)

The body of analytical and computational methods by which characteristics of a population are inferred through the observations made in a sample of that population is called statistics. The population here is the large set of data from the process plant.

1.1.4 Definitions

Fault: "A fault is an unpermitted deviation of at least one characteristic property (feature) of the system from the acceptable, usual, standard condition."

Disturbance: undesired and uncontrollable interference acting on the system

Fault detection: Finding if there is any fault in the system and also the time of the fault.

Fault Isolation: Determining the type and location of the fault.

Fault identification: Determining the magnitude and time-variant behavior of the fault.

Fault diagnosis: "Fault diagnosis is determining which fault has occurred, in other words, determining the cause of the observed out-of-control status".

Monitoring: Observing and recording the progress of different variables in a process over a period of time.

Error: An error is the deviation of the measured value from the actual or true value.

Failure: "A failure is a permanent interruption of a system's ability to perform a required function under specified operating conditions."

Malfunction: "A malfunction is an intermittent irregularity in the fulfillment of a system's desired function."

1.2 COLLOCATION OF FAULTS

Faults that occur in a process plant can be classified into three types:

1.2.1 Sensor Faults

These are due to erroneous reading from the sensors. This can be due to broken wires, lost contact with the surface, etc. (these come under complete sensor faults) in which case the reading shown by the sensor is not related to the value of the measured physical parameter. This can also

be due to gain reduction, biased measurement or increased noise (these come under partial sensor faults), in which case the reading still contains useful information which can be retrieved.

1.2.2 Actuator Faults

Just like the sensor faults, even these can be classified into partial or complete actuator fault (loss of control action). When there is a breakage, burned or cut wires, then in-spite of the input applied to an actuator, no actuation is produced. This can be categorized under complete actuator fault. In case of partially failed actuator only part of the normal actuation is produced. It can be a result of hydraulic or pneumatic leakage, reduced input voltage or increased resistance.

1.2.3 Component Faults

All those faults that cannot be categorized under sensor or actuator faults comes under component faults. These faults usually occur due to structural damages of the components. The dynamical behavior of the system changes as a result of these faults. Among different type of faults, component faults are the most commonly encountered ones.

1.3 FAULT DETECTION

Once a fault has occurred in a system, the immediate step will be detecting the fault. These faults can be detected using different methods. Fault detection methods can be classified into following:

1.3.1 Data-Driven Methods

Data driven methods use only the availability of large amount of data for many industrial process. This data can be transformed and used as a knowledge bank using many different ways.

This process is also called as feature extraction. This feature extraction can be done using statistical or non-statistical methods. The important class of non-statistical is neural networks. Statistical feature extraction methods mainly contain Principal Component Analysis (PCA), Fisher Discriminant Analysis (FDA), Partial Least Squares (PLS) and statistical pattern classifiers. The only drawback of data-driven methods is that it largely depends on the quality and quantity of the process data.

1.3.2 Model-Based or Analytical Methods

There are different model based fault detection methods. Fault detection using input and output measurements of the system is the basic one among them. In some cases we measure only the output signal, in such a case model based methods such as spectral analysis and band-pass filters are used for fault detection. Among these methods, the most frequently used techniques are parameter estimation and observer based methods. Fault detection using these methods is done by comparing the system's measured variables with the information obtained from the system's mathematical model.

1.3.3 Knowledge-Based Methods

Fault detection using knowledge based methods is a heuristic process. System characteristic values like amplitude, variance, state variables, model parameters and vibrations are used to extract features during normal and faulty conditions by using the heuristic and analytical knowledge. After extracting the features under both the conditions (faulty and normal), they are then compared and methods of change detection are applied. Artificial neural networks, fuzzy logic and neuro-fuzzy based can be regarded as knowledge-based methods.

1.4 FAULT IDENTIFICATION

Fault identification is to determine the process variables that are responsible for the fault and thus the information can be used to diagnose the fault. It will be helpful to focus on the subsystems where the fault occurred in a large plant. It not only helps to diagnose the fault more efficiently but also saves time and money.

1.5 FAULT DIAGNOSIS

The whole process of determining the kind, size, location of the fault including fault isolation and identification is called as fault diagnosis. Based on the observed analytical and heuristic symptoms and the heuristic knowledge of the process, the diagnostic procedure is outlined. The knowledge from physical laws or quantitative measurements and observations is analytical diagnostic knowledge. The knowledge that is not clearly written or described, but is obtained as a result of learning through experimental methods is heuristic diagnostic knowledge.

1.6 STRUCTURE OF THIS THESIS

In this thesis only part of the fault isolation is addressed, i.e. fault detection, specifically MATLAB codes were written for the PCA algorithm and Simulink models were built to take the data to implement the PCA algorithm for fault detection.

In chapter 2 the basics of the fault detection and some background about the process control has been discussed.

In chapter 3 the principal component analysis (PCA) algorithm was explained in detailed and then tested on two problems, one with a random data set (two dimensional data) considering

two classes and the other by taking data from a linear first order process (three dimensional data) built in MATLAB Simulink.

In chapter 4 data from the Tennessee Eastman (TE) challenge problem is taken and the PCA algorithm built was tested on this data (41 dimensional data) and then the results using two different methods of fault detection, with two different types of faults are compared and a case study of one of the fault is done in detail.

In chapter 5 conclusions are drawn and the possible extensions of the thesis are presented.

CHAPTER 2

FAULT DETECTION AND PROCESS CONTROL

2.1 BACKGROUND AND LITERATURE REVIEW

Fault detection is an important process in process control. Abnormal event management (AEM) mainly depends on timely fault detection and diagnosis of the process faults. The petrochemical industries lose an estimated $20 billion every year in the United States alone [14]. Modern plants are more difficult to control and diagnose. This is because of many factors like large scale complex configurations, plant wide integration and fewer experienced operating personnel. Process control has made lot of advances in the last three decades with the advances in the computer technology. Many control operations such as opening and closing values, which are low level control actions and called as regulatory control were earlier performed by human operators, but now are operated using automated equipment with the aid of computers. However, more important control tasks are still operated by humans. These tasks include responding to abnormal events in the process control plant. This involves detecting the faults in time and identifying the cause for the fault, diagnosing that cause and then bringing back to a normal and safe operating state. This entire activity is called as Abnormal Event Management (AEM).

However, completely relying on human operators to cope with such abnormal events has become increasingly difficult due to many factors. Due to the broad scope of diagnostic activity that comprehensively includes a variety of malfunctions such as parameter drifts, process unit failures, process unit degradation, and so on. Furthermore it is complicated by the size and complexity of modern process control plants. For instance, there may be as many as 1500

process variables in a large process control plant observed every few seconds (Bailey 1984) leading to information overload.

With such conditions, it's more obvious that human operators tend to make incorrect decisions and take decisions that make the matter even more worse. About 70% of industrial accidents are caused due to human errors [14]. These abnormal events have significant effect on safety, environmental and economic impact. In-spite of advanced computer based control of chemical plants. There are two worst accidents that occurred in the chemical plants. Union carbide's Bhopal, India, accident and Occidental Petroleum's Piper Alpha accident. One more major accident is the explosion at the Kuwait Petrochemical's Mina Al-Ahmedi refinery in June 2000 as a result of which around 100 million dollars are lost in damages [14].

Hence, this has become a challenge for the control engineers. So, having fully reliable methods to control has become an important task on hands. The present challenge is that of automating the AEM with the aid of intelligent control system. Hence assistance can be provided for the human operators where ever there is a need. This is viewed as the next major milestone in control systems research by the persons in the process control industry.

The first step of AEM is the automation of process fault detection. There are different approaches to fault detection, one based on modeling of the system's process and the other based on data history of the system's process. Here we are going to use the past data of the process to build a PCA model for fault detection. The entire fault isolation process using system data is shown in Figure 2.1. First the data is extracted from the system and then the noise effects are reduced using dimensionality reduction technique (here PCA is used). This reduced dimensional data is compared with the system data under normal operating conditions to detect the presence of fault. Then next step after fault detection is to place and process variables responsible for the

fault (fault identification), once the fault identification is done, appropriate corrective action has to be taken in order to diagnose the fault.

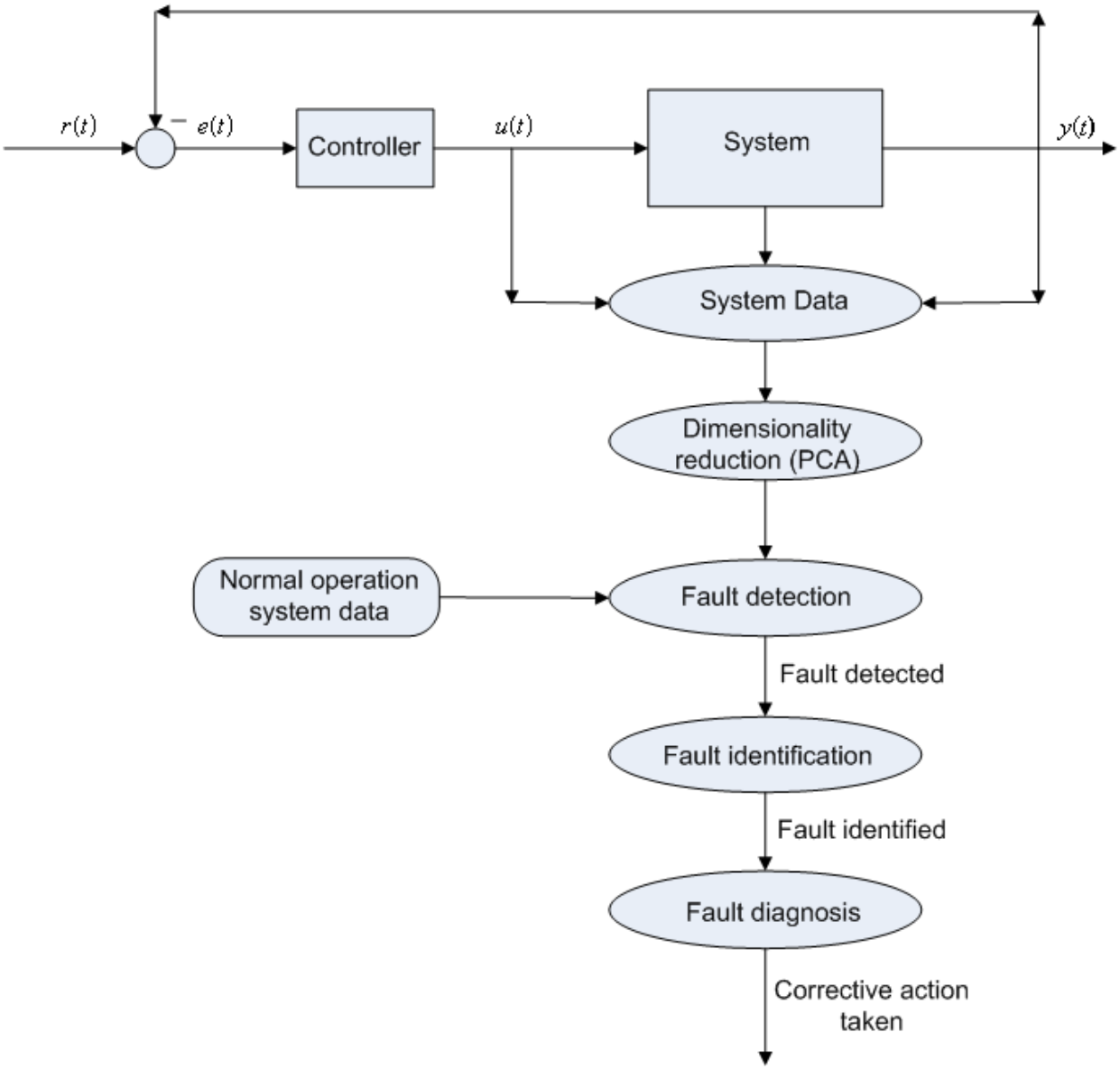

Figure 2.1 Block diagram showing the entire fault isolation process

Data driven techniques are broadly classified into two types, qualitative and quantitative. The classification of data driven methods is shown in detail in Figure 2.2. Qualitative analysis of the data is done either by using expert systems or by analyzing the trends in the data. Quantitative information can be extracted from the past data using different methods. These can be broadly classified into two; non-statistical methods and statistical methods. The major component of non-statistical method of feature extraction is neural networks and the major components of the statistical methods are Principle Component Analysis (PCA), Partial Least Squares (PLS) and other statistical methods for feature extraction.

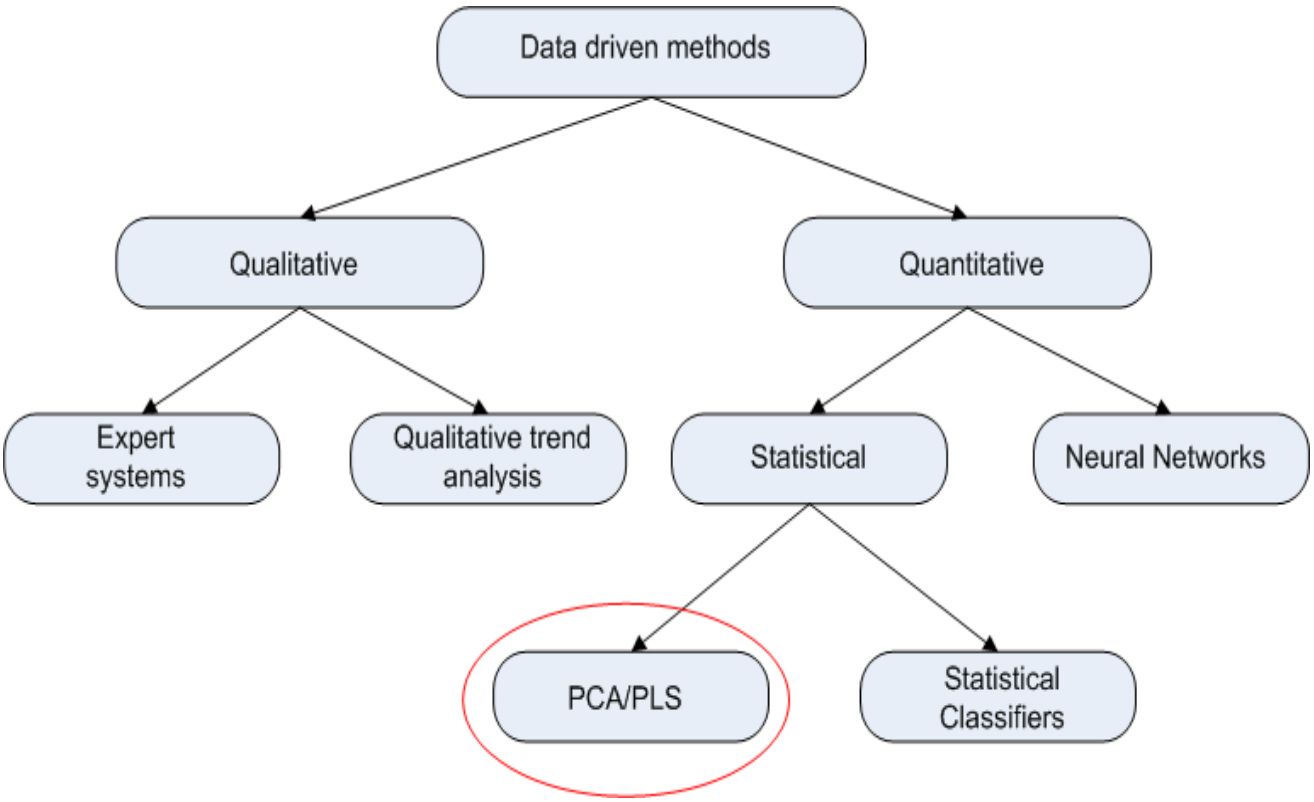

Figure 2.2 Pyramid of data driven methods

2.2 MULTIVARIATE STATISTICAL METHODS

Multivariate statistical techniques, one of the effective tools, are useful in compressing data and reducing its dimensionality. They retain essential information that is easier to analyze than the original huge data set, and they are also known to handle noise and correlation to extract true information effectively. A standard multivariate technique called as PCA method was initially proposed by Pearson [9] and later developed by Hotelling [10]. This method has been included in many textbooks (Anderson, 1984; Jackson, 1991) as well as research papers (Wold, 1978; Wold, Esbensen, & Geladi, 1987). Transforming a number of related process variables to a smaller set of uncorrelated variables is one of the key functions of multivariate statistical techniques.

2.3 PROCESS INDUSTRY

In terms of process control and process industry, a process is defined as a sequence of steps that changes or refines raw materials to produce final products. During the process the raw materials which pass through are measured, mixed, heated or cooled, filtered, or dealt with some way or the other to get the final product. The oil and gas industry, the food and beverage industry, the pharmaceutical industry, steel industry, power industry, chemical industry and water treatment industry all comes under process industries.

2.3.1 Process Control

Process control refers to the different approaches used to control process variables when a product is being manufactured. For instance, aspects like proportion of the compounds, how well the compounds are blended; temperature of the materials, pressure under which the material

is processed can impact the quality of the final product significantly. The production process is controlled for three reasons:

(i) Minimize Variance

To ensure a consistent high-quality final product, variance in the final product has to be minimized. Minimizing variance can also save money for the manufactures. For instance, in blending process of gasoline, a number of different components are combined to get a mixture meeting the specifications of a particular grade. If the flow control of different components is not accurate, the percentage of components with high-octane may be in excess or inadequate in gasoline. As a result, the customers would receive a higher grade at lower price or lower grade at higher price, which is a loss, to the company in the first case and to the customer in the second. Minimizing the variance can also save money by avoiding the need of *padding*. *Padding* is the process of making a product of higher quality than it needs to be according to the specifications. When the process control is poor, manufacturers has to pad the final product to make sure that the specifications are met, which adds to the cost. So, with a more accurate process control the optimal point can be set closer to the actual product specifications, thus cutting the cost down.

(ii) Maximize Efficiency

In some processes maximum efficiency can be achieved by maintaining the process at a particular condition. For instance, in a hydroelectric power plant, generator has maximum efficiency at 90% of valve opening. In a chemical plant, the production temperature has to be maintained at a particular value to ensure process efficiency. Hence, accurate process control also helps in maximizing the efficiency of the process.

(iii) Ensure Safety

Safety is also important in any process. An out-of-control nuclear or chemical process can be catastrophic. So it is very essential to maintain accurate control of the all the process variables. For instance, for maintaining the desired state of fission reaction in a nuclear reactor the control rods are to be inserted at a particular level, without which the fission reaction goes uncontrollable which is clearly a threat to not only workers in the plant but also to the people living around the plant. Hence, accurate process control is required to ensure proper safety.

2.3.2 Process Control Flow Charts (PCFC)

These are very important for uninterrupted quality control of the plant. The main advantage of using these flow charts in the process control plants is that we can see the exact time when the fault has occurred. During the time the process is observed, these charts can be used to observe the fluctuations in a variable that is being measured. These are simple looking charts with a setoff points plotted as a two dimensional plot, with time along x-axis and variable measurement along y-axis, that are connected to form a smooth curve. The average and the threshold measurements are represented with additional horizontal lines through these curves. So, if any variable measurement is out of these limits, it can be shown on the chart. Hence, these charts can be used for detecting the faults in a system.

For example if you say that the measurement of a variable is a sine curve with unit magnitude, then the threshold values will be +1 and -1 and the average would be 0. So we can say that if a measurement exceeds the threshold, a fault has occurred in the system. In the Figure

2.3 we can see that the plot in red crosses the threshold values, whereas the plot in green is well within the thresholds and the plot in blue just touches the threshold value but doesn't cross the threshold value. So from Figure 2.3 we can say that the plot in red in under faulty conditions.

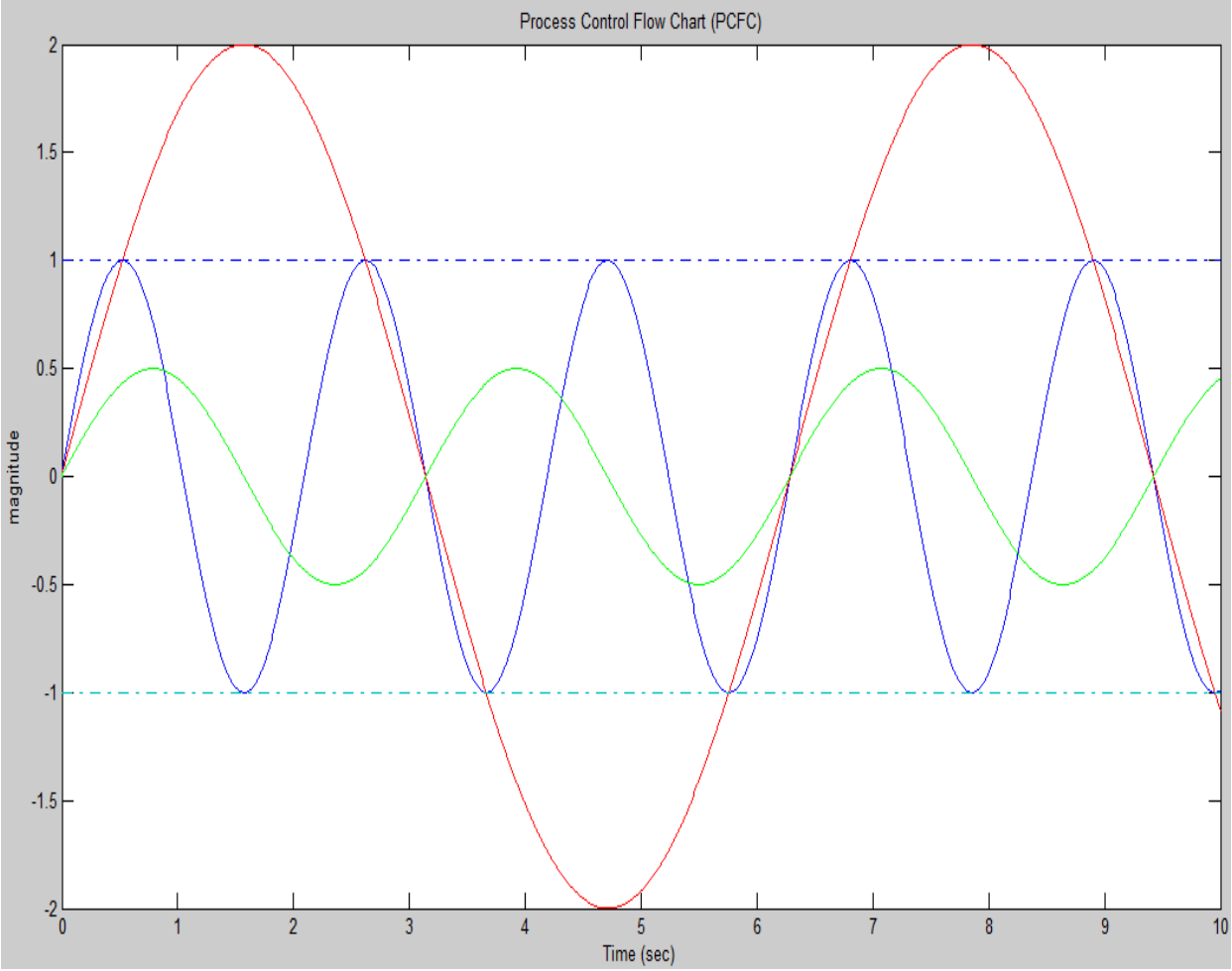

Figure 2.3 An example of PCFC **(**Simple sine waves showing thresholds).

CHAPTER 3

PRINCIPAL COMPONENT ANALYSIS (PCA)

3.1 BACKGROUND AND LITERATURE REVIEW

Principal Component Analysis (PCA) is a dimensionality reduction technique without much loss of information [1]. Though the idea of PCA appeared in 1889, it's research and applications are still in study. It is a statistical technique that is greatly used in face recognition, image compression, and also used for finding pattern in high dimension data. The principle components are determined using the raw data. There are many uses of the principle components found; interest may be of first principle component or 2^{nd} to Nth principle component or the first few principle components.

Motivation for application of PCA:

1. Lower dimensional data representations can be produced using PCA, thus eliminating the noise effects, and therefore increasing the efficiency of fault detection and diagnosis.
2. The structure derived from PCA is helpful in identifying the variables responsible for the fault.

The dimensionality of a data set having a large number of interrelated variables can be reduced using PCA without loss of the variation present within the data set. Process control plants have large amount of data taken during normal operating (no fault) conditions, this data is taken and is transformed into new set of variables called principal components, this set of new variables are uncorrelated.

3.2 PCA METHODOLOGY

The original n-dimensional data matrix will be transformed into a lower dimension without any loss of variance between the variables. A set of orthogonal vectors called loading vectors are determined using the variance between the variables. The methodology guarantees that the variance of the data will be very large in the direction of some eigenvectors called Principal Components (PC) and much smaller in the direction of other eigenvectors, so the variance of the data can be approximated to the PC neglecting the other eigenvectors. When we have two or three dimensional data the number of principal components can be found by just looking at the eigenvalues of the covariance matrix 'C' and when we have huge we can use the following methods to find the principal components.

1. Parallel Analysis
2. SCREE procedure
3. Cumulative Percent Variance (CPV) approach
4. Cross validation.

3.2.1 PCA Algorithm

Step 1: The raw data is arranged in the form of a matrix X with each column representing a different variable that is being measured. For example if there are m variables and each being measured n times then the matrix $X \in \Re^{n x m}$.

$$X = \left[x_1(k)\ \ x_2(k)\ \ x_3(k)\ \ \ldots\ldots\ \ x_m(k)\right],\ k = 1, 2, 3\ \ \ldots\ldots\ n \tag{3.1}$$

This data matrix is then scaled in order to avoid the possibility of particular variables dominating inappropriately in the procedure of reducing the dimensionality. Scaling is done by

first mean centering the data as our objective is to capture the variation of the data from the mean. Let each column of the mean centered matrix (P) be p_i and it is given by

$$p_i = x_i - \bar{x}_i \quad , i=1,2,.....,m \tag{3.2}$$

Then each column of the mean centered matrix P is divided by the corresponding column's standard deviation. This is given by

$$p_i^{std} = \frac{p_i}{\sigma_{p_i}} \tag{3.3}$$

where p_i^{std}, for $i=1,2,3,....n$ are the columns of the updated data matrix P_{std}. Each column of this matrix now has zero mean and unit variance.

Step 2: Calculating the covariance matrix C.

The covariance matrix is given by

$$C = \frac{(P'_{std})^* P_{std}}{n-1} \tag{3.4}$$

Step 3: Finding the loading vectors by using the Singular Value Decomposition (SVD)

$$C = \frac{(P'_{std})^* P_{std}}{n-1} = USV' \tag{3.5}$$

Where the diagonal matrix S contains the non-negative real eigenvalues arranged in the descending order of their magnitude. In order to minimize the effect of noise loading vectors corresponding to only the largest a eigenvalues are to be considered.

Step 4: Finding the value of a

This can be done by using any of the methods that were mentioned in the Section 3.2.

Here parallel analysis method is used which is considered as most reliable method [5] among all those different methods. (More about parallel analysis (PA) and the code used for getting the value of a is being provided in appendix C). Once the value of a (which signifies the number of principal components to be considered) is found from the plot of real data eigenvalues and percentile random eigenvalues, the loading vectors corresponding to the first a eigenvalues are selected from the loading matrix V .

The matrix thus formed by the first a columns of the V matrix is represented by L

$$L = V(1:m, 1:a) \tag{3.6}$$

Step 5: Calculating the Score Matrix.

The **score matrix** T is calculated using

$$T = L' X \tag{3.7}$$

Step 6: Back transformation of score matrix into reduced dimensional observation space.

The transformation is done using the formula,

$$\hat{X} = LT \tag{3.8}$$

The data matrix now has minimal noise effects.

Step 7: Fault detection in new data set

The new data set under faulty condition is given by the matrix X_{fault}.

$$T_{fault} = L' X_{fault} \tag{3.9}$$

$$\hat{X}_{fault} = L T_{fault} \tag{3.10}$$

The fault in the data is calculated by plotting the error between the two data sets $\hat{X}_{fault}$ and $\hat{X}$

$$e = \hat{X}_{fault} - \hat{X} \tag{3.11}$$

3.3 IMPLEMENTATION OF PCA

3.3.1 Example 1: A random set of two dimensional data (classes were considered)

The application of PCA algorithm is shown by taking a random set of data. The data set consists of two classes containing 2 measurements and with number of observations equal to 10. This is done to just give an overview and visualize as to what is being done by the PCA algorithm. It reduces the noise by first calculating a set of orthogonal vectors called loading vectors as mentioned earlier and then selecting only the loading vectors corresponding to the a (largest) singular values. Since the example here is 2-D one, we can find the value of a by looking at the eigenvalues of the covariance matrix. The covariance matrix is decomposed (using SVD) to get,

$$S = \begin{bmatrix} 1.2331 & 0 \\ 0 & 0 \end{bmatrix}$$

So, the eigenvalues of the covariance matrix are $\lambda_1 = 1.2331$ and $\lambda_2 = 0$. The largest value being 1.2331, only the loading vector corresponding to that is considered. The dimensionality of the data is now reduced to 1-D. The loading vector matrix V is calculated to be,

$$V = \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{bmatrix}$$

Since $a = 1$, we will consider only the first column of the loading vector matrix V to form the score matrix T.

In Figure 3.1, the red and black dots represent classes1 and 2 of the original data set respectively. After applying the PCA algorithm, we can see that the whole data is along a straight line (green-class1 and pink-class2 dots), which implies the noise effects have been reduced and the whole data is now along a straight line (one dimension).

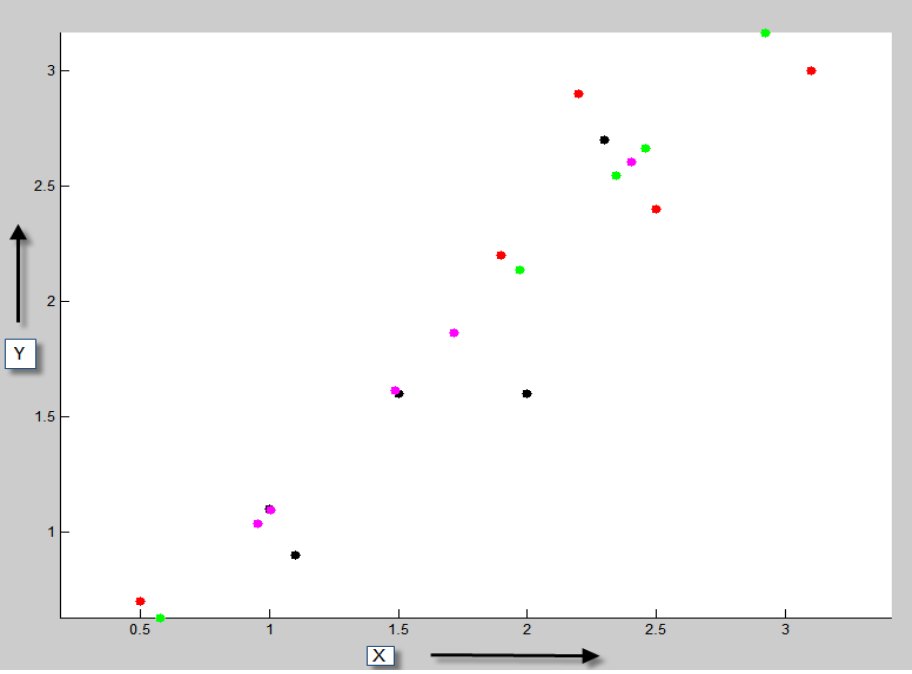

Figure 3.1 PCA applied to random two dimensional data

3.3.2 Example 2: A three dimensional data set generated using a linear first order process model built in MATLAB Simulink.

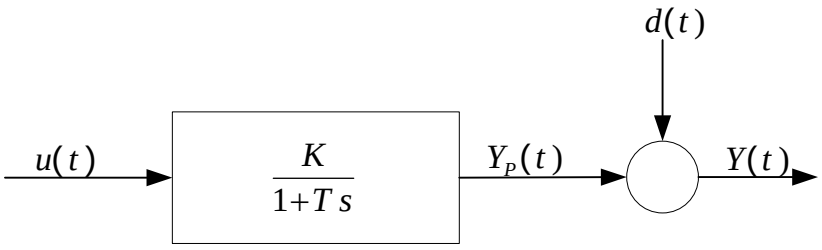

Figure 3.2 Plant Model (linear first order process) used for data extraction

Where, $K = 10, T = 10$ sec, $u_0 = 2$ and $\omega = 3$

The input is given by $u(t) = u_0 \cos(\omega t)$. The output data matrix X is given by

$$X = \begin{bmatrix} u & \dot{Y} & Y \end{bmatrix} = \begin{bmatrix} u(0) & \dot{Y}(0) & Y(0) \\ .. & .. & .. \\ .. & .. & .. \\ .. & .. & .. \\ u(N) & \dot{Y}(N) & Y(N) \end{bmatrix} \tag{3.12}$$

In this example, the data is taken from a linear first order system shown in Figure 2.1. The model is simulated and the output data is used in the code to implement PCA. The results are shown in Figure 3.3.

The covariance matrix is decomposed (using SVD) to get,

$$S = \begin{bmatrix} 102.9353 & 0 & 0 \\ 0 & 4.4909 & 0 \\ 0 & 0 & 0.2676 \end{bmatrix}$$

So, only the first two eigenvalues are considered, as the third eigenvalue is clearly very small compared to the other two.

The loading vector matrix is calculated to be,

$$V = \begin{bmatrix} -0.0160 & 0.9999 & 0.0049 \\ -0.9986 & -0.0157 & -0.05 \\ -0.0499 & -0.0057 & 0.9987 \end{bmatrix}$$

Here a = 2, so only the first two columns of the above matrix are considered to form the score matrix T.

In Figure 3.3(a), we can see the original data, which is all over the 3-dimensional space. After applying the PCA algorithm taking two principal components we get the data as shown in Figure 3.3 (b), where we can see that the data (with reduced noise effects) is along a single plane.

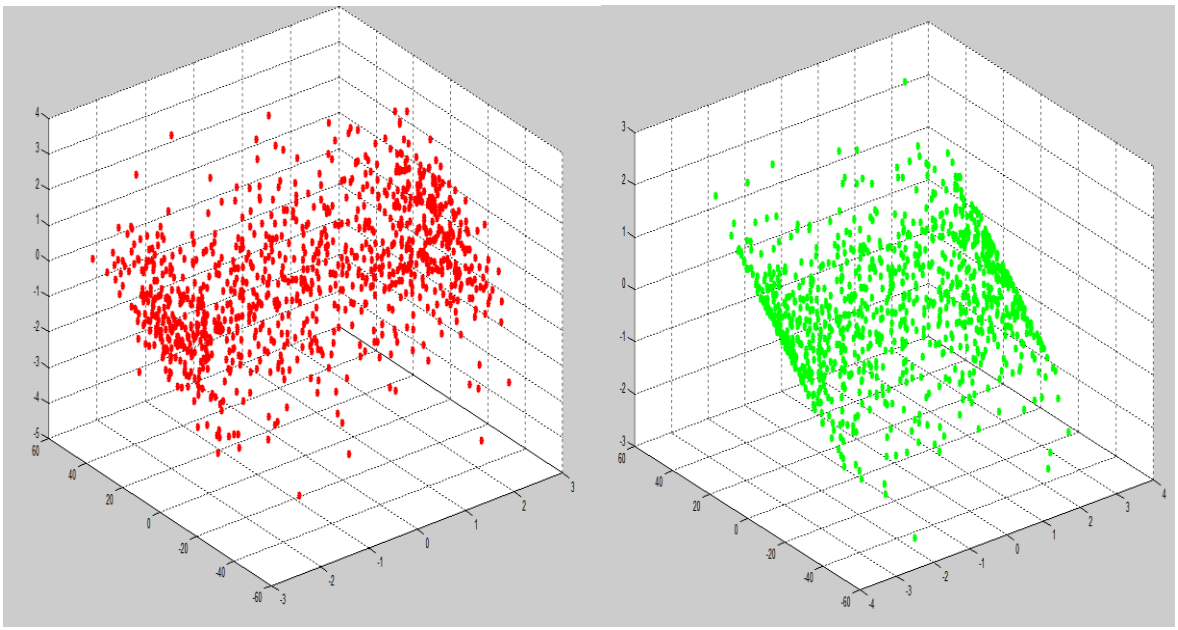

Figure 3.3(a) Original data extracted from the linear first order process (b) Data after PCA analysis

Now, white noise is added at the input of the Simulink model and the output data is collected. Applying the PCA model developed to this data gives same distribution as shown in Figure 3.4, but we can see that the data looks stretched (dots shown in blue) compared to the data under normal operating conditions (dots shown in green). All the points that are outside the thresholds (lines shown in red) shown in the Figure 3.4 are considered to be faults.

However, while modeling PCA for a real plant, we get huge data and the observation variables will be more than three, so it is not possible the whole data set in a single plot. In that case, the error between the values of each variable under normal operating conditions and faulty conditions ($e = \hat{X}_{fault} - \hat{X}$) is plotted verses time for detecting the faults. There are other methods that can be used to detect the faults that are discussed in the next section

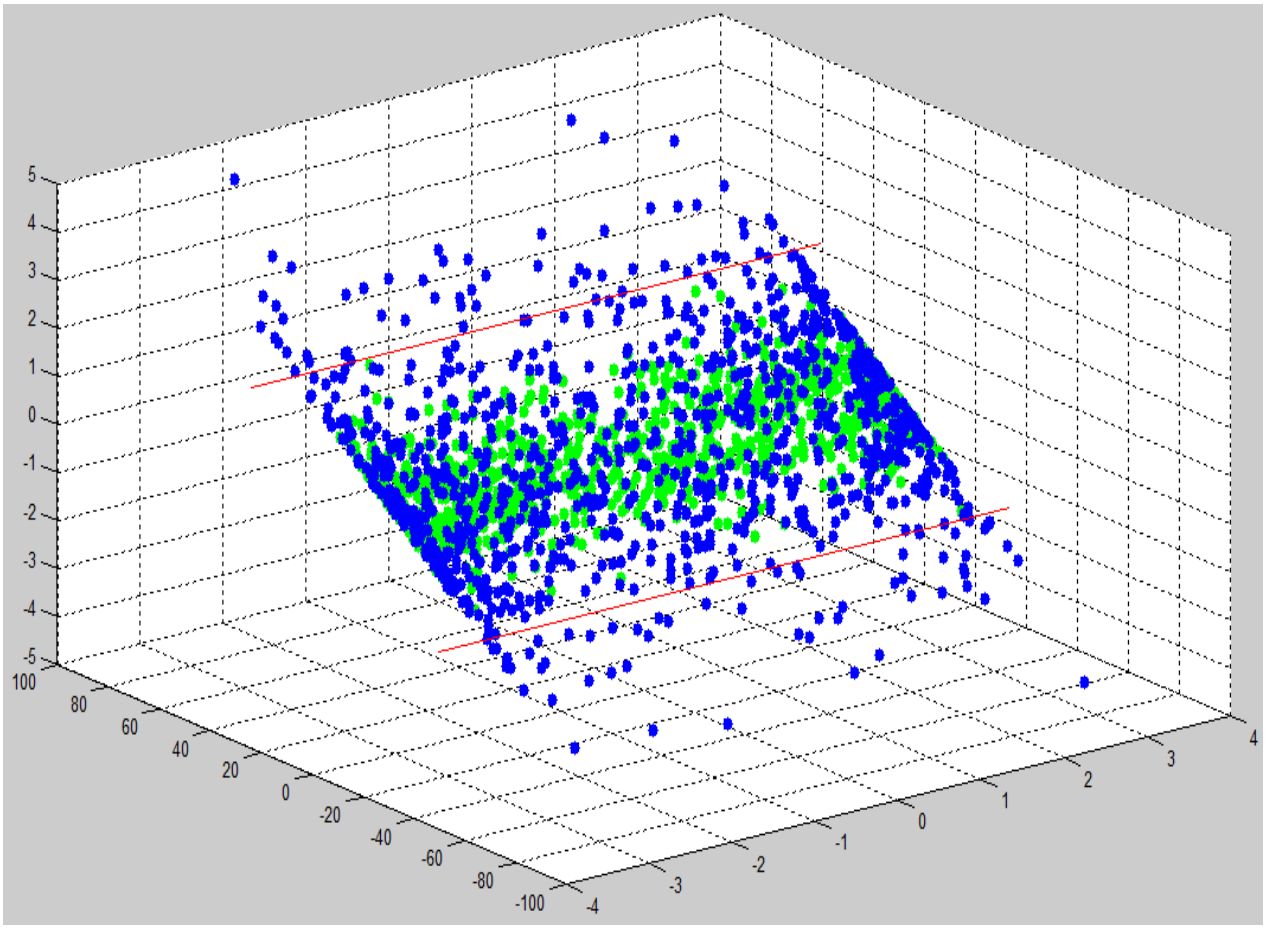

Figure 3.4 Data after PCA analysis of the plant (under normal (GREEN) and faulty (BLUE) operating conditions)

3.4 FAULT DETECTION USING PCA

The faults in a system can detected after applying PCA algorithm to the by any of the following methods.

Change detection of

1. T Matrix.
2. Back transformed $\hat{X}_{fault}$.
3. Comparing X and $\hat{X}_{fault}$.

Significant fluctuations in the variable $X_i(k)$ can be detected, if it exceeds a certain threshold value using any of the above methods.

Fault detection by observing the limits (thresholds). Under normal conditions the output Y should be in the range, $Y_{\min} < Y < Y_{\max}$. So when this condition is not satisfied by certain output, i.e. the output $Y(t)$ exceeds any threshold, then we can say that there is a fault in the system. The selection of threshold values should be in such a way that the false alarms are avoided and the faults are properly detected. So, there will be a tradeoff between very small bandwidth and very large bandwidth of the thresholds.

CHAPTER 4

IMPLEMENTATION OF PCA ON TENNESSEE EASTMAN PROCESS

4.1 TENNESSEE EASTMAN (TE) PROCESS

Most of the people working in the field of process control thought that it would be very helpful for the design purpose that if they can test the models built for process control on real time system data. So the Eastman Chemical Company provided the Simulink model for testing the process control and monitoring methods [11]. The Tennessee Eastman process is the most widely used example in the process control industry to get data for testing different methods of process control. It is a realistic problem for testing the PCA algorithm. The Simulink model of the TE process is available at http://depts.washington.edu/control/LARRY/TE/download.html.

The flow sheet of the Tennessee Eastman process from Downs and Vogel [11] is shown in Figure 4.1.

Since the process is just used to get the output data and use to build a PCA model for fault detection, the process is not discussed in detailed here .However, the nonlinear state variable form of the model is as follows,

$$\dot{x} = \frac{dx}{dt} = f(x, u, d) \tag{4.1}$$

$$y = g(x, u, d) \tag{4.2}$$

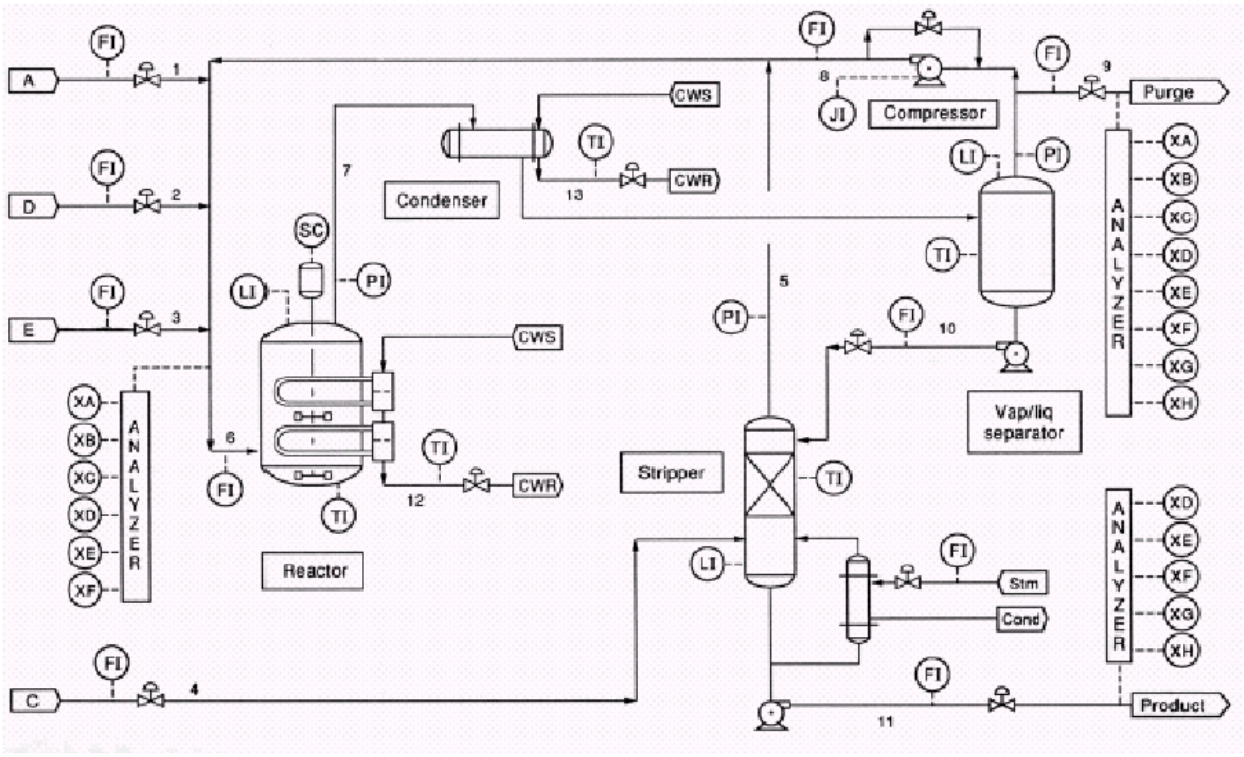

Figure 4.1 Flow sheet of the Tennessee Eastman Process

The input (u) to the process is a vector of length 10 and the output (y) is a vector of length 42 and the unmeasured inputs are represented with the disturbance vector (d). The elements of the d vector are described in the Table 4.1.

The complete process and state equations are available in the reference [13].

Table 4.1 Faults for Tennessee Eastman in the Simulink model

Fault ID	Description	Type
IDV1	A/C feed ratio, B composition constant (Stream 4)	Step change
IDV2	B composition , A/C ratio constant (Stream 4)	Step change
IDV3	D feed temperature	Step change
IDV4	Reactor cooling water inlet temperature	Step change

(Cont. Table 4.1)

IDV5	Condenser cooling water inlet temperature	Step change
IDV6	A feed loss (Stream 1)	Step change
IDV7	C header pressure loss – reduced availability (Stream 4)	Step change
IDV8	A,B,C feed composition (Stream 4)	Random change
IDV9	D feed temperature (Stream 2)	Random change
IDV10	C feed temperature (Stream 4)	Random change
IDV11	Reactor cooling water inlet temperature	Random change
IDV12	Condenser cooling water inlet temperature	Random change
IDV13	Reaction kinetics	Slow drift
IDV14	Reactor cooling water valve	Sticking
IDV15	Condenser cooling water valve	Sticking
IDV16-20	Unknown	
IDV21	Valve for stream 4 fixed at the steady state position	Constant Position

4.2 SIMULINK MODEL AND CODE

The downloaded code consists of a C-program and a Simulink program that has the whole TE process with in the S-function block. The S-function block consists of a mex-file that has to build using the C-program before attempting to run the Simulink model.

The data is taken from a Simulink model (S-function) representing the Tennessee Eastman (TE) challenge process. Here the S-function is written in C-language and it will be used in the Simulink in the form of a C-mex file. C-mex file is built after compiling the C file containing the S-function (temex.c) in Matlab using the command “mex temex.c”. This will

create the mex file in the same folder containing the S-function file (temex.c). The Simulink model is the run with all the values for disturbance vector to be equal to zero.

The initial conditions are taken as defined in the Downs and Vogel paper. There are 12 input signals as defined by Downs and Vogel. These can be manipulated by the user. The output consists of 41measured variables. These are sent to the Matlab workspace, so that they can be plotted once the simulation is done and also that the data can be used later to build the PCA model.

A detailed explanation about the Mex-files and also about the system requirements for creating a Mex-file from the C-program are provided in appendix A.

The disturbance block circled in red in the Figure 4.2 consists of a vector of length 20. When the value of each element in the vector is equal to 0 then that means there is no disturbance in the system (normal operating conditions), otherwise there is a disturbance in the system. For our case we are assuming the disturbance to be a fault, which is not true in the real sense (disturbance is different from a fault).

The total number of outputs in this simulation is 41 defined by Downs and Vogel. Among these 22 outputs are measured continuously and the remaining are sampled composition analyses from chromatographs.

4.3 MATLAB SIMULATIONS

The simulation is run for 48 hours and the data sets are generated applying each of the 21 faults after a simulation time of 10 hours and one set under normal operating conditions (fault 0). The PCA model is trained by using the data set under normal operating conditions. Once the model is trained it is now tested with the remaining 21 data sets. The number of principal

components is calculated using parallel analysis. Only the real data eigenvalues greater than the percentile random eigenvalues are considered, the value here is found to be 11 and the same is shown in Figure 4.3

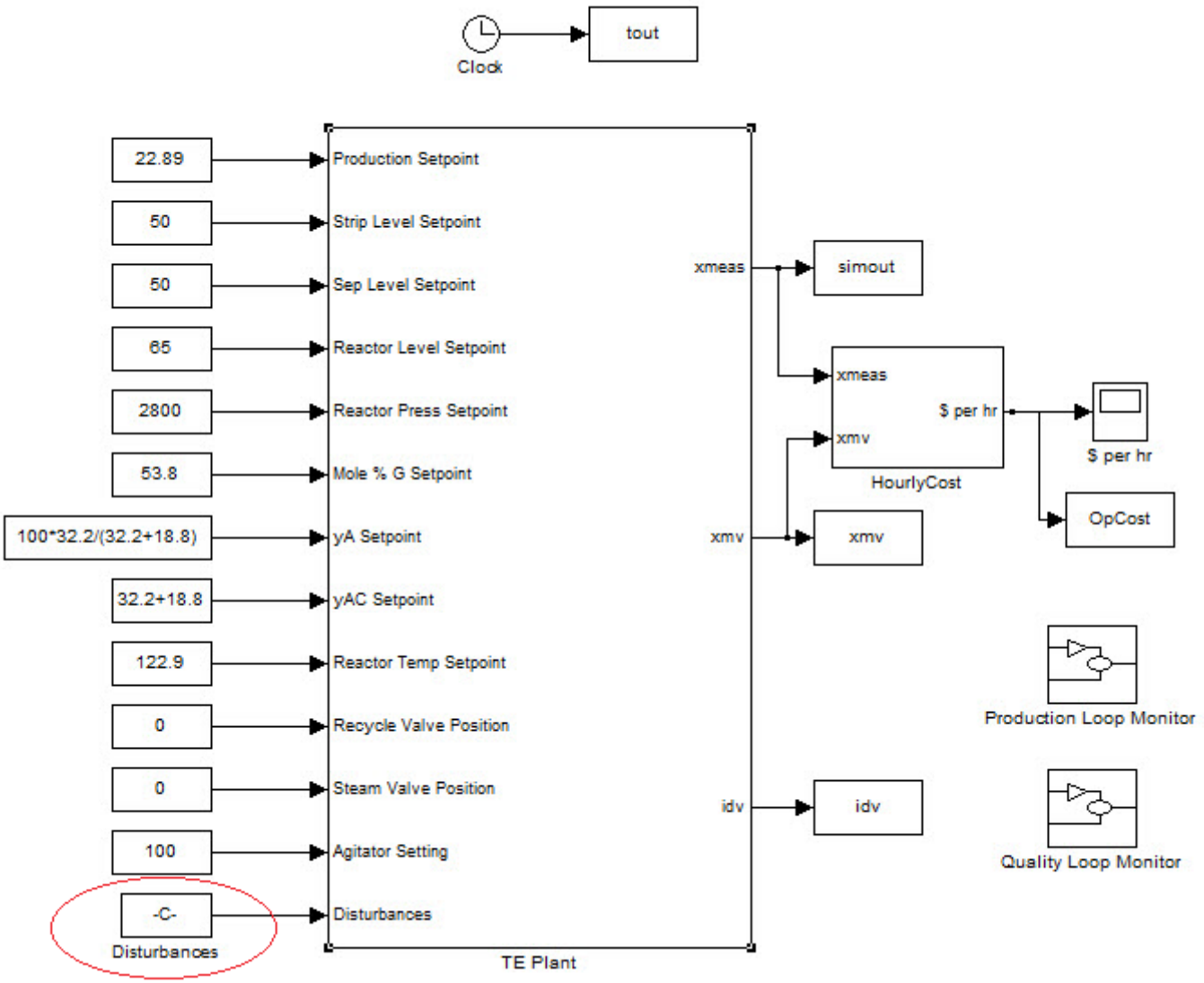

Figure 4.2 Simulink model of the TE challenge process

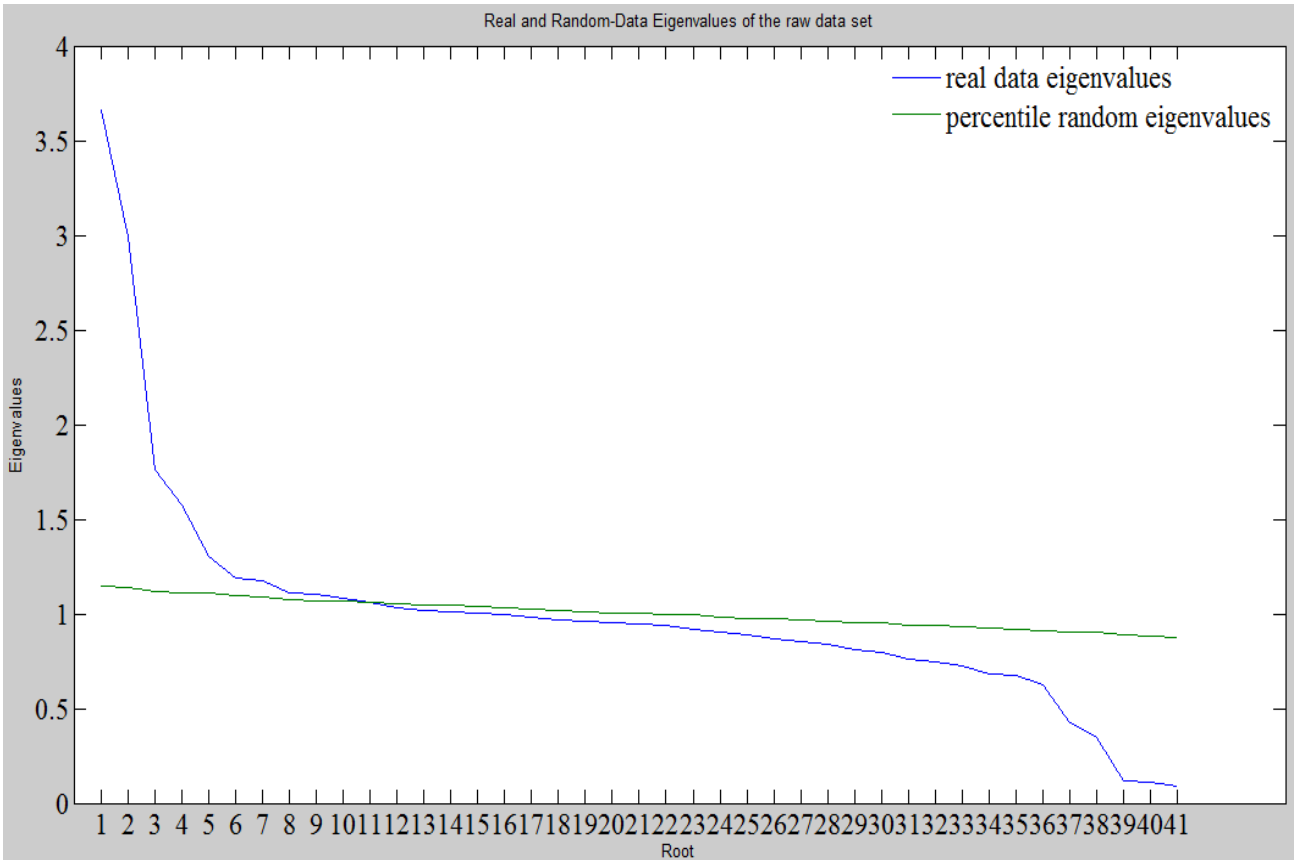

Figure 4.3 parallel analysis plot for the TE process data

4.3.1 Fault Detection

The presence of faults is detected using two ways,

1. By directly looking at the output plots from the simulation using Process Control Flow Charts (PCFC) as discussed in Section 2.3.2 and
2. By taking the output data from the simulation and apply PCA model that was build.

The results from the direct simulation are compared with the results from the PCA model. When the first method was used, sometimes a considerable amount of change was not observed in the outputs when there is a particular fault (i.e. the measured value lies wirh in the threshold even when there is a fault). The observations of few of the outputs are tabulated in

Table 4.2. The observations are made for six different faults (IDV1, IDV2, IDV4, IDV5, IDV11, and IDV12). The 'X' mark in the box implies that the particular fault does not have a considerable effect on that particular output. For example, the fault IDV2 does not show considerable change in the reactor level output.

Table 4.2 List of some of the faults that were unnoticed in certain outputs

	Faults					
Outputs	IDV1	IDV2	IDV7	IDV8	IDV11	IDV13
D feed(2)	X	X	X	X	X	
Reactor level(8)		X	X		X	
Reactor Temp(9)	X	X		X		
Stripper underflow(17)	X	X	X	X	X	
Stripper Steam Flow(19)	X	X	X	X	X	X
Component C to reactor(25)		X			X	
Component D to reactor(26)		X	X	X	X	X
Component E to reactor(27)		X		X	X	
Component C in purge(31)		X			X	
Component D in purge(32)	X	X	X	X	X	X
Component D in product(37)	X	X		X	X	X
Component F in product(39)	X			X	X	
Component G in product(40)	X	X				
Component H in product(41)		X		X		

On the other hand, when the second method (Fault detection using PCA) is used even a small change is detected immediately without much delay.

In the first method, the fault is detected when the measured value goes outside the threshold value. When there is fault it was observed that not all the output plots cross the threshold values immediately, as a result some delay was observed in some of the cases. Actual fault was introduced after 10 hours of simulation time, so with an ideal time delay of zero seconds the fault should be detected at t=10hrs. However, this is not the case in real and there is bound to be a delay time. The delay time is calculated by subtracting 10 from time when the reading crosses the threshold value (as the fault is introduced into the system after 10 hours). These delay times are tabulated in the table below. The results are tabulated for the six faults used in Table 4.1 and several outputs are considered.

Table 4.3 Delay times during fault detection using both the methods (time is in hours)

Outputs	IDV1		IDV2		IDV7		IDV8		IDV11		IDV13	
	1	2	1	2	1	2	1	2	1	2	1	2
D feed(2)	4.4	3.4	9.7	5.2	-	12.1	20.0	12.1	21.3	12.2	12.5	7.2
Reactor level(8)	3.0	1.0	-	3.4	-	10.3	11.2	10.3	-	-	3.2	1.7
Stripper underflow(17)	3.5	2.7	-	4.3	-	-	20.0	19.9	-	-	4.5	2.1
Stripper Steam Flow(19)	-	3.4	-	3.4	-	-	-	12.6	-	10	-	2.1
Component D to reactor(26)	0.91	0.12	-	4.3	-	0.02	6.8	3.9	15.1	5.1	7.3	1.4
Component E to reactor(27)	2.0	1.8	-	3.4	-	-	8.4	7.9	-	-	2.8	1.4

(Cont. Table 4.3)

Component D in purge(32)	1.0	0.9	-	4.8	9.4	1.0	15.8	10.3	-	11.6	9.2	3.9

From Table 4.3 it is evident that applying PCA to the raw data greatly reduces the delay times, for example when we take the case of fault IDV13 (Reaction kinetics), the fault is detected after 12.5 hours in the D-feed reading when we use the first method, because only after 12.5 hours the reading crosses the threshold value. On the other hand when the second method is used the delay time is reduced to 7.2 hours. Some of the results are as shown in Figures 4.4-4.7.

In Figures 4.4-4.7 the plot in BLUE is under normal operating conditions and the plot in RED is under the fault that is specified below the figure. The thresholds are calculated separately for plots (a) and (b). The original data under normal operating conditions is plotted (in BLUE) and the upper and lower limits of this data are taken as thresholds for the plot (a). Then the PCA algorithm was applied to the original data under normal operating conditions and this reduced dimensional data is plotted (in BLUE) and the upper and lower limits of this data are taken as thresholds for the plot (b). Once the thresholds are calculated, the fault is introduced and the original data under fault is plotted (in RED) in plot (a) and then the data after applying the PCA algorithm is plotted (in RED) in plot (b).

4.3.2 Case Study on Fault IDV1 (A/C Feed Ratio, B Composition Constant (Stream 4))

The advantages of using the PCA algorithm over crude way of observing the process variables are discussed in detail by taking in this case study of the fault IDV1. The fault introduces a step change in the A/C feed ratio in stream 4 shown in Figure 4.1. Say, the feed ratio is decreased, so this results in a decrease in A-feed and an increase in the C-feed.

This results in a decrease in the A-feed in the recycle stream 5 and the controller in the plant reacts to increase the A-feed in stream 1 which can be seen in Figure 4.8. These two effects cancel out each other after sometime and this result in constant A-feed composition in stream 6 after sometime as shown in Figure 4.9.

"The variation in the flow rates and compositions of stream 6 to the reactor causes variations in reactor level, which affects the flow rate in stream 4 through a cascade control loop, as shown in Figure 4.10. The flow rate of stream 4 eventually settles to a steady-state value lower than its value at normal operating conditions."

Since the ratio of the reactants A and C changes, the distribution of the variables associated with the material balances (i.e. level, pressure and composition) changes correspondingly. Since more than half of the variables monitored deviate significantly from their normal operating behavior, this fault is expected to be easily detected. Process monitoring statistics that show poor performance on Fault 1 are likely to perform poorly on the other faults as well". The last sentence here can be supported with the observation made in Table 4.1. We can clearly see that if the fault IDV1 is missed in a particular output, then even the other faults are also missed in most of the cases.

4.4 DETECTING MULTIPLE FAULTS

When we have multiple faults it is sometimes difficult to detect the fault, because the effect of one fault may be cancelled by the other and the fault goes undetected leading to a catastrophic event. So, we need more sophisticated methods to detect the faults when we have multiple faults at the same time. However, when we are sure that the effect of one fault does not cancel the other, then we can use the same method (PCA) for fault detection.

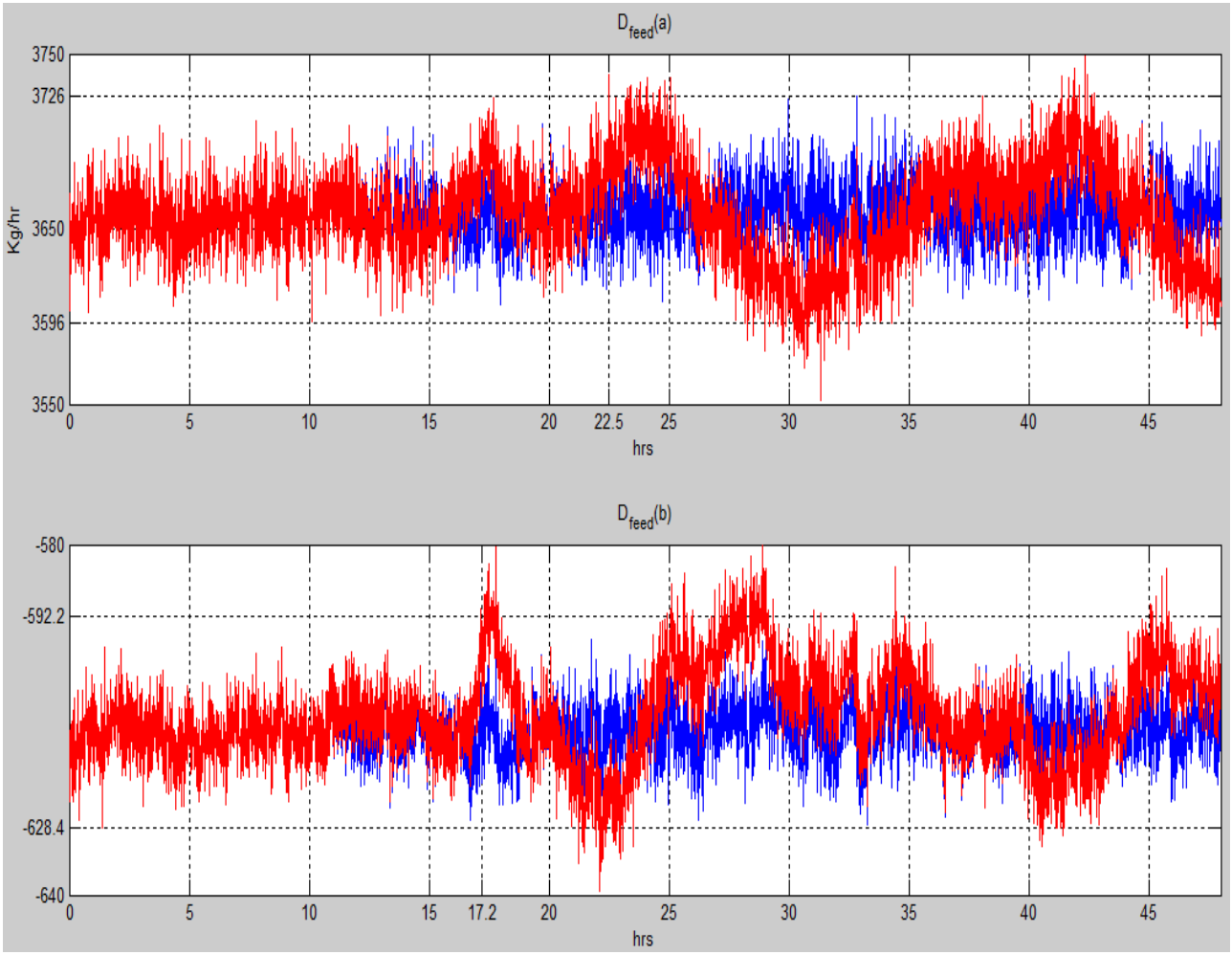

Figure 4.4 Plot showing the delay time using the two methods - (a) Original data (under normal operation-BLUE, under fault (IDV13)-RED) and (b) Data after applying PCA algorithm(under normal operation-BLUE, under fault (IDV13)-RED)

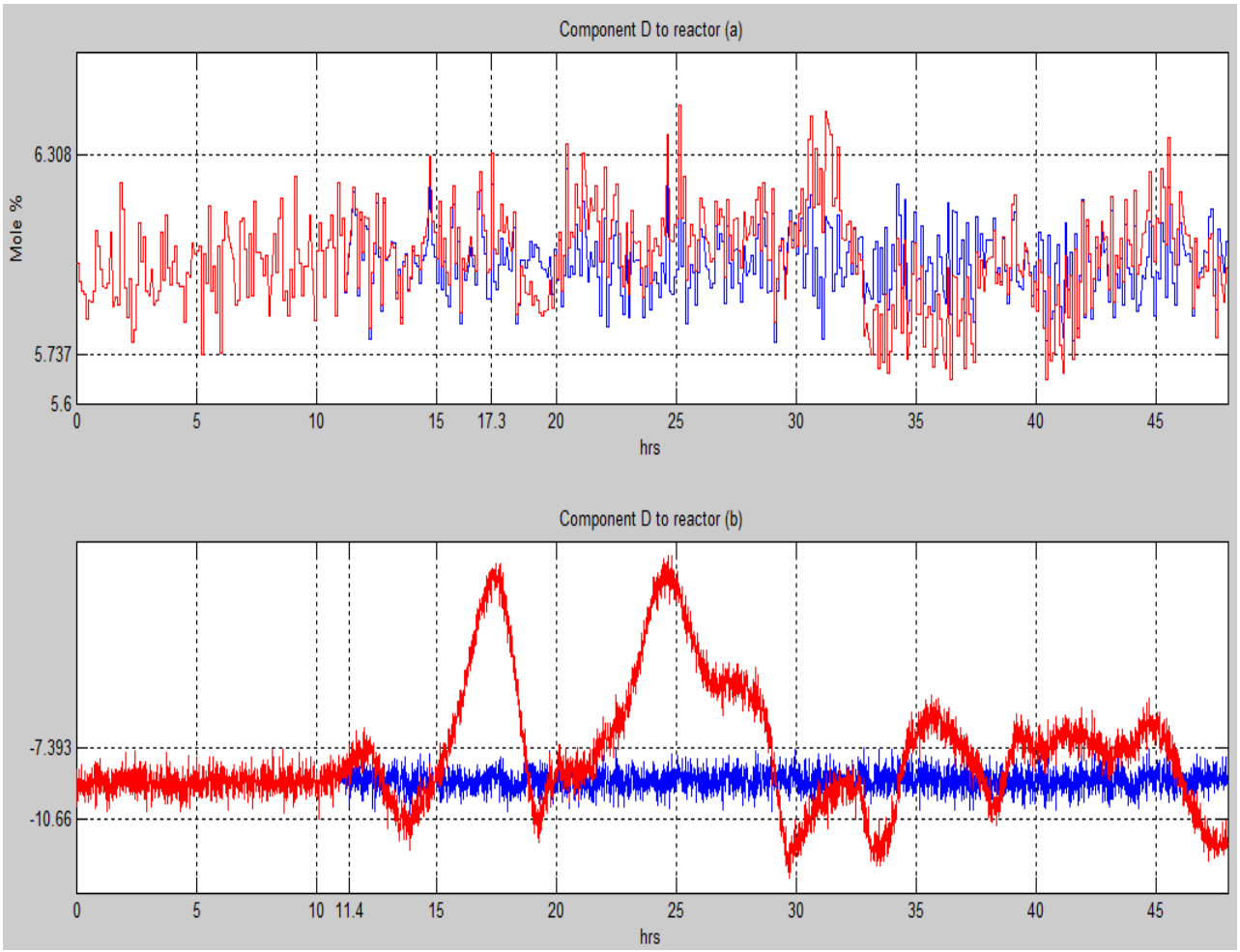

Figure 4.5 Plot showing the delay time using the two methods - (a) Original data (under normal operation-BLUE, under fault (IDV13)-RED) and (b) Data after applying PCA algorithm(under normal operation-BLUE, under fault (IDV13)-RED)

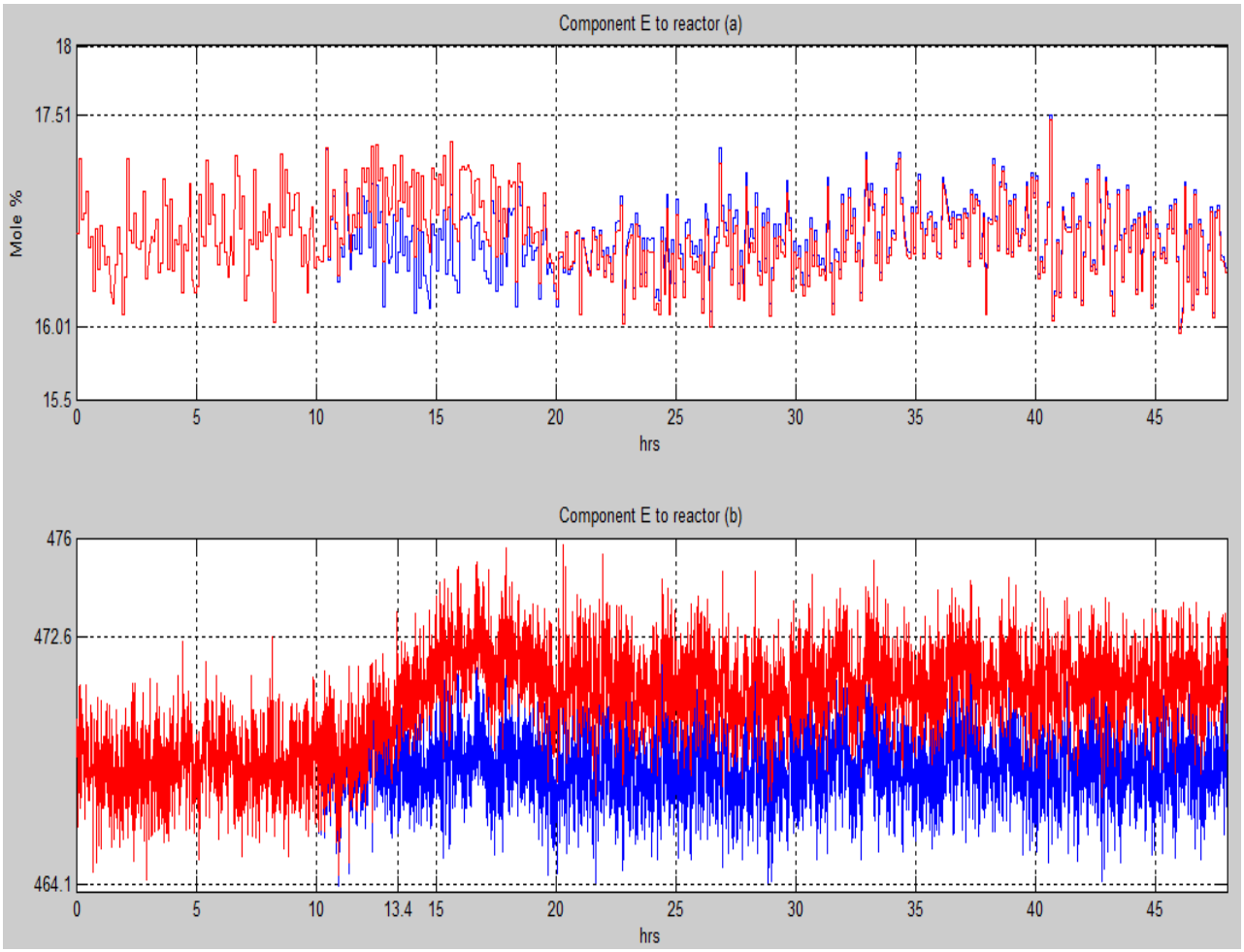

Figure 4.6 Plot showing the delay time using the two methods - (a) Original data (under normal operation-BLUE, under fault (IDV2)-RED) and (b) Data after applying PCA algorithm(under normal operation-BLUE, under fault (IDV13)-RED)

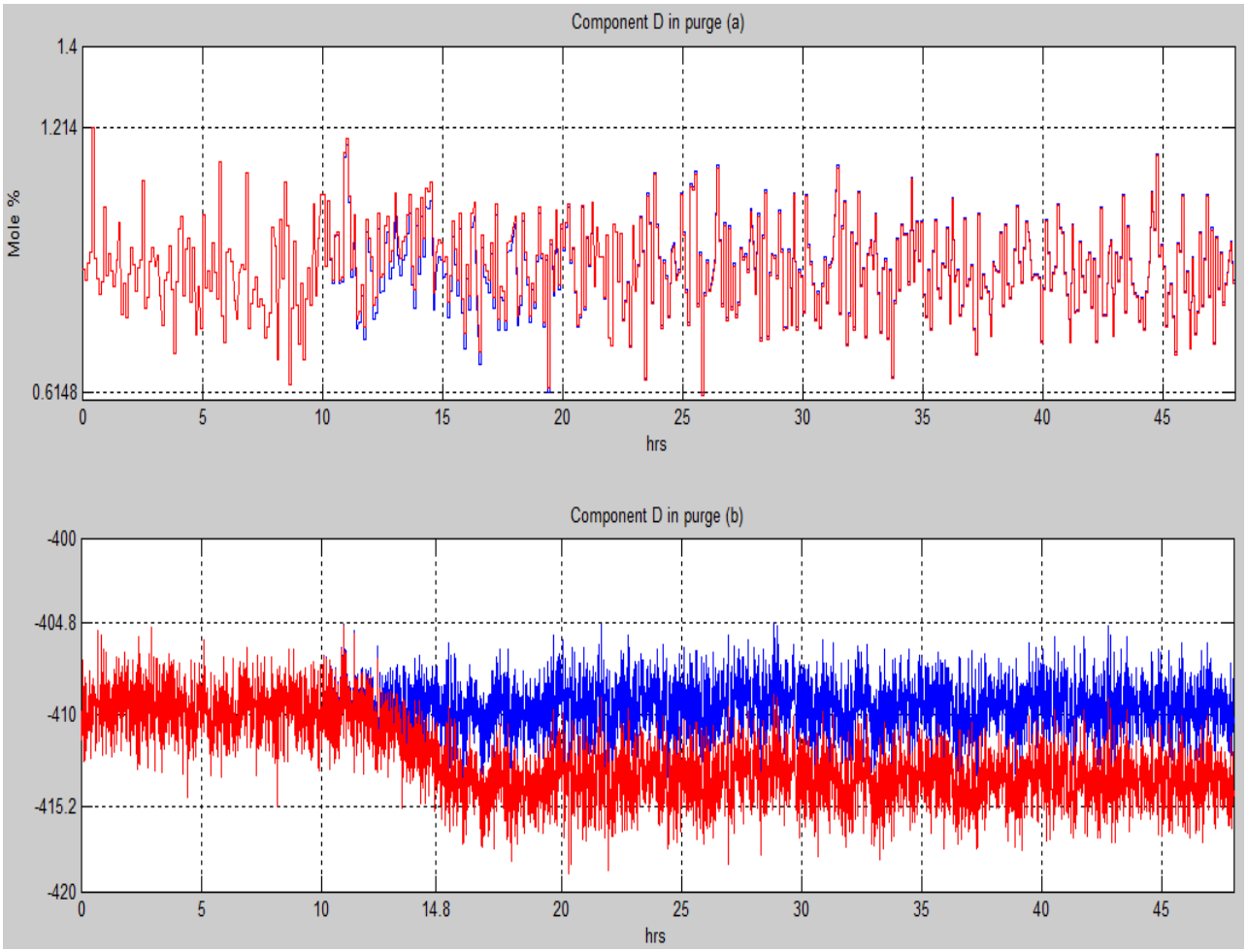

Figure 4.7 Plot showing the delay time using the two methods - (a) Original data (under normal operation-BLUE, under fault (IDV2)-RED) and (b) Data after applying PCA algorithm(under normal operation-BLUE, under fault (IDV13)-RED)

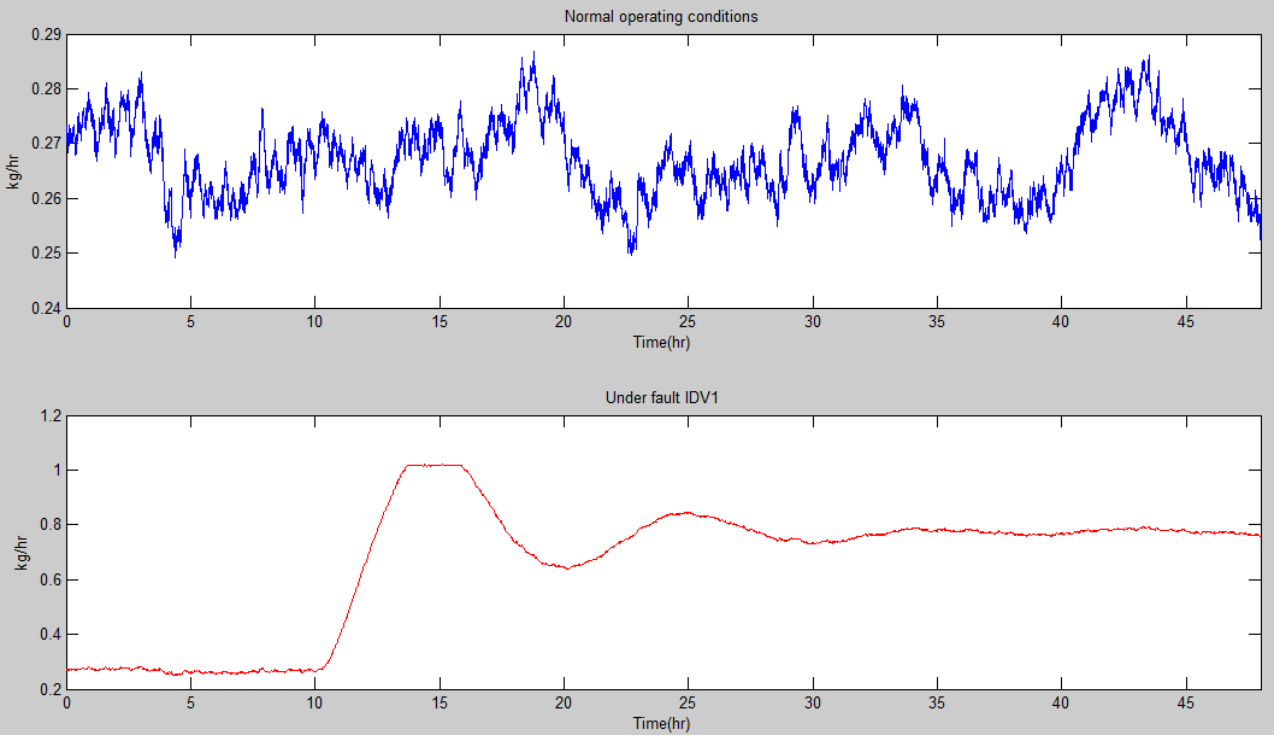

Figure 4.8 Outputs of A-feed under normal and fault (IDV1) conditions

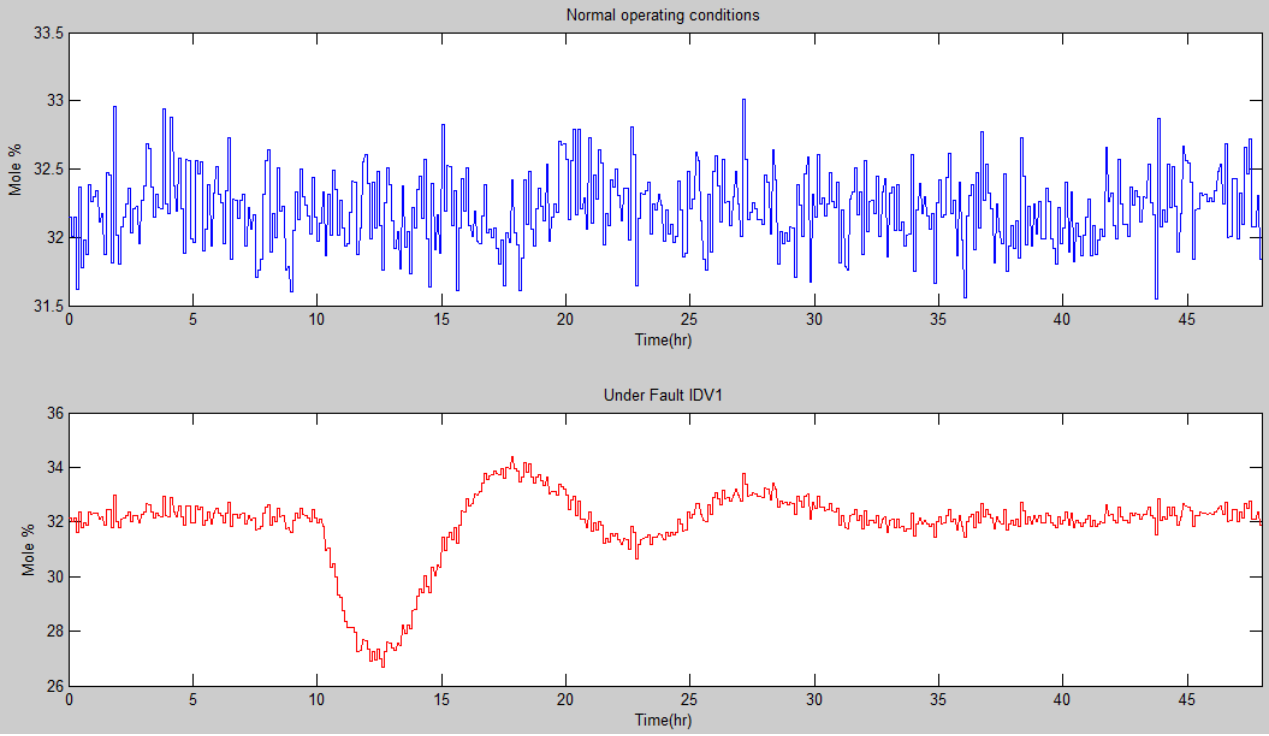

Figure 4.9 Outputs of component A to reactor (Output 23) under normal and fault (IDV1) conditions

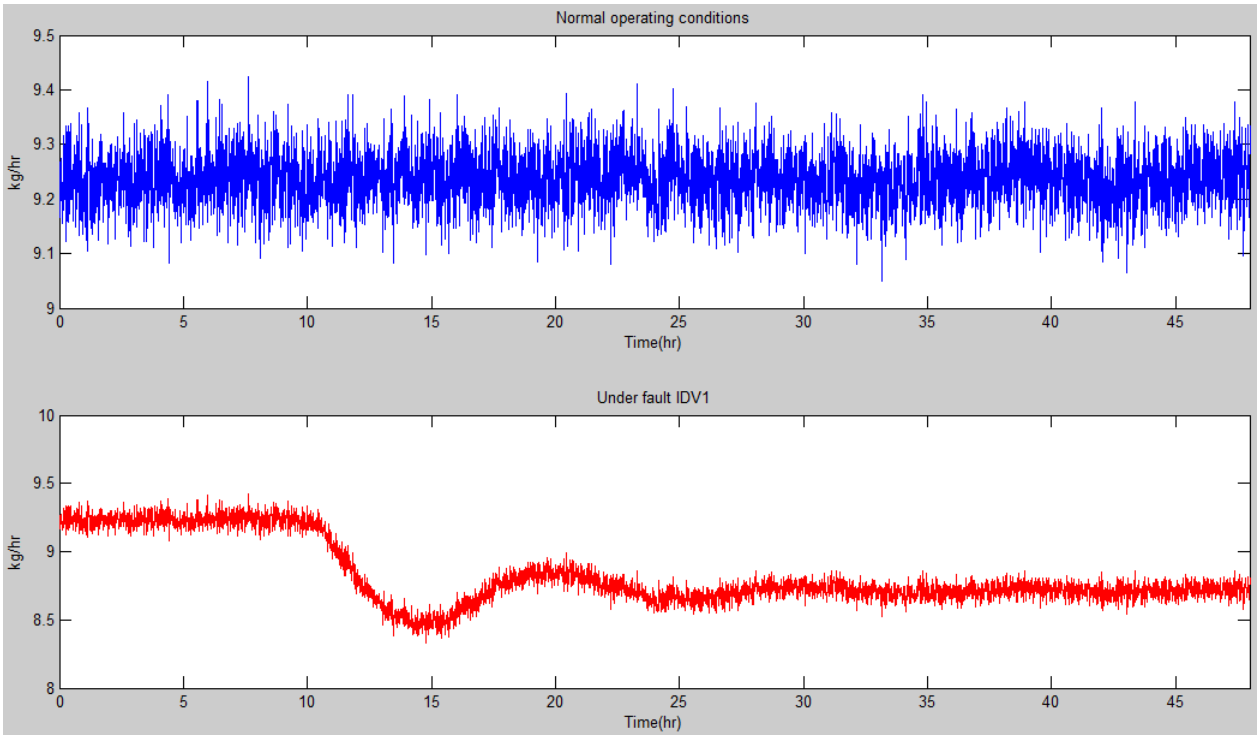

Figure 4.10 Outputs of A and C feed (Output-4) under normal and fault (IDV1) conditions

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSIONS

The mission of fault detection is to take the huge data from the archives and determine when the fault has occurred. The fault can be associated with equipment failure, wear and tear of equipment or extreme process faults. The accuracy of detecting the faults using process data can be improved by using data dimensionality reduction techniques like Principal Component Analysis (PCA), Fisher Discriminant Analysis (FDA) and Partial Least Squares (PLS), etc. The PCA has been studied in detail and the advantages it has over the crude way of just fixing the threshold values for a measurement and observing it directly. The number of undetected faults is largely reduced when the dimensionality technique was used rather than direct observation of the variables. It is observed that the delay times are also greatly reduced when the PCA algorithm was used. So, by applying the dimensionality reduction techniques to raw data the sensitivity to all possible faults of the process is greatly increased and the promptness to the detection of the faults is also improved substantially.

The execution times of the program were found to be as shown in the Table 5.1

Table 5.1 Program Execution times

Program	Execution time
Two dimensional data	0.20 sec
Three dimensional data	0.25 sec
TE challenge problem data	50 sec

From the table we can clearly see the trivial observation that the execution time is directly proportional to the amount of data. A much better algorithm is needed for a real time on-line fault detection because sometimes even a slight delay (here we are talking about time to execute the code) has to avoided for proper functioning of the system.

5.2 FUTURE WORK

As discussed earlier, Fault isolation includes three steps

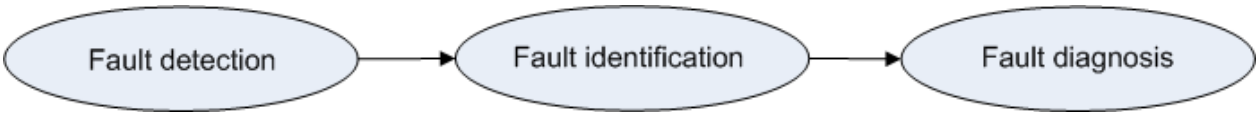

Figure 5.1 Fault isolation procedure

The complete process of fault isolation includes three steps as shown in Figure 5.1; however in this thesis I discussed only about the first step, fault detection. Once we detect that there is a fault in the system, the next step would be to find the exact kind and place of fault (Fault identification) in the system, which can be done using other multivariate statistical methods like FDA, PLS, etc. Finally the fault has to be diagnosed by taking the proper corrective action depending on the type of fault. The problem can be extended to the next part of fault identification and classifying them under different classes. FDA can be used for this purpose.

.

REFERENCES

[1] E. L. Russell, L. H. Chiang, R. D. Braatz, "Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis," *Chemometrics and Intelligent Laboratory Systems* 51, 81–93(2000).

[2] L.H. Chiang, E. L. Russell, R. D. Braatz, "Fault Detection and Diagnosis in Industrial Systems," *Springer*, London (2001).

[3] R. Isermann, "Fault-Diagnosis Systems An Introduction from Fault Detection to Fault Tolerance," *Springer*, London (2006).

[4] J.J. Downs, E.F. Vogel, "Plant-wide industrial process problem control" *Comput. Chem. Eng.* Vol 17, pp. 245-255 (1993).

[5] "Introduction to statistical process control techniques," *Statit software, Inc*., Corvallis, Oregon.

[6] K. R. Skinner, D. C. Montgomery, G. C. Runger, J. W. Fowler, D. R. McCarville, T. R. Rhoads, J. D. Stanley, "Multivariate Statistical Methods for Modeling and Analysis of Wafer Probe Test Data," *IEEE transactions on semiconductor manufacturing*, vol. 15, no. 4, November 2002.

[7] A. Delorme, "Statistical methods," *Swartz Center for Computational Neuroscience, INC*, University of San Diego, CA.

[8] S. B. Franklin, D. J. Gibson, P. A. Robertson, J. T. Pohlmann, J. S. Fralish, "Parallel Analysis: a method for determining significant principal components"

[9] K. Pearson, "On lines and planes of close fit to systems of points in space," *The London, Edinburgh and Dublin Philosophical Magazine and Journal*, vol. 6, Issue 2(1901)

[10] H. Hotelling, "Multivariate Quality Control Illustrated by Air Testing of Sample Bombsights," *Techniques of Statistical Analysis*. New York: McGraw-Hill (1947).

[11] W. Svante, K. Esbensen, P. Geladi. "Principal component analysis." *Chemometrics and Intelligent Laboratory Systems* 2(1):37–52(1987).

[12] PAControl.com, *Instrumentation & control*, process control fundamentals.

[13] Jockenhovel, Biegler and Wachter, Comp.Chem.Eng., 27, 1513-1531, 2003.

[14] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis part I: Quantitative modelbased methods," *Computers and Chem. Eng*. 27, 293—311 (2003).

[15] Y. Qingsong, "Model-based and data driven fault diagnosis methods with applications to process monitoring," PhD. thesis, Case Western reserve University, May 2004.

[16] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis part III: Process history base methods," *Computers and Chem. Eng*. 27, 327—346 (2003).

[17] K. Erdal, "Fault detection and diagnosis in nonlinear dynamical systems," Middle East Technical University, PhD. thesis, August 2005.

[18] J. P. George, Z. Chen, P. Shaw, "Fault Detection of Drinking Water Treatment Process Using PCA and Hotelling's T2 Chart," *World Academy of Science, Engineering and Technology* 50 (2009).

[19] T. Villegas , M. J. Fuente, M. Rodríguez , "Principal Component Analysis for Fault Detection and Diagnosis. Experience with a pilot plant," *Advances in Computational Intelligence, Man-Machine Systems and Cybernetics* 147-52.

[20] L. I. Smith, "A tutorial on Principal Components Analysis," February 2002

[21] M. N. Nashalji, M. A. Shoorehdeli, M. Teshnehlab, "Fault Detection of the Tennessee Eastman Process Using Improved PCA and Neural Classifier," *International Journal of Electrical & Computer Sciences IJECS* Vol: 9 No: 9

[22] A. Mauricio Sales Cruz, "Tennessee Eastman Plant-wide Industrial Process Challenge Problem Complete Model," Technical University of Denmark, January 2004

[23] J. C. Hayton, D. G. Allen, V. Scarpello, "Factor Retention Decisions in Exploratory Factor Analysis: a Tutorial on Parallel Analysis," *Organizational Research Methods*, 7, 191(2004).

[24] B. P. O'Connor , "SPSS and SAS programs for determining the number of components using parallel analysis and Velicer's MAP test," *Behavior Research Methods, Instruments, & Computers, 32 (3), 396-402(2000)*

APPENDIX A

MEX-FILES

A.1 INTRODUCTION TO MEX-FILES

MATLAB has lot of built-in functions, but not all functions we want to use will be built-in. Sometimes code for the functions to be used will be already written in C, C++ or FORTRAN. In order to use these functions along with the built-in functions, we would need to interface these external functions to MATLAB so that these functions could be called from MATLAB as if they were MATLAB built-in functions.

MEX stands for MATLAB executable. These are the dynamically linked programs written in C, C++ or FORTRAN which after compiling can be used directly from MATLAB. The functionality to exchange data between MEX-files and MATLAB is provided by the external interface functions. MEX-files are written for two main reasons,

1. The functions written inside large existing C, C++ or FORTRAN source code can be call directly from MATLAB without having to rewrite the entire code as MATLAB files.
2. The other major advantage is speed. The computational time is reduced by a substantial amount.

The code in a MEX-file consists of two different parts

1. Code (computational routine) to perform all the computations that a function has to perform.
2. Code (gateway routine) that interfaces the computational routine with MATLAB.

A.2 EXAMPLE OF THE MEX-FILE

Every MEX-file includes `mex.h`. Thus, the MEX code starts like this:

```
#include "mex.h"
```

Every MEX-file should have the `mexFunction` entry point. The code now becomes

```
#include "mex.h"
void mexFunction(int nlhs, mxArray *plhs[],
    int nrhs, const mxArray *prhs[]) {
```

Then an API function is added to make the MEX-file do the required operation. Finally the code becomes

```
#include "mex.h"
void mexFunction(int nlhs, mxArray *plhs[],
    int nrhs, const mxArray *prhs[]) {
  mexPrintf("Hello, world!\n");
}
```

A.3 INSTALLATION INSTRUCTIONS

The system should have a C-compiler that is compatible with the version of MATLAB that is installed in the system. Once a C-compiler is installed, the compiler has to be set from within the MATLAB. The procedure is explained in the following steps:

1. Open MATLAB and type “mex –setup” and it show the following message

   Would you like mex to locate installed compilers [y]/n?
2. Press “y” and enter, if the installed compiler is compatible with the version of MATLAB, the window displays the different compiler options available. For example, with

MATLAB R2008 a and Microsoft C++ 2008 express compiler it should show the following message

Select a compiler:

[1] Microsoft Visual C++ 2008 in C:\Program Files (x86)\Microsoft Visual Studio 9.0

[0] None

Compiler:

3. Select the compiler you want to use by entering the appropriate number. For example entering 1 here would display the following message

   Please verify your choices:

   Compiler: Microsoft Visual C++ 2008

   Location: C:\Program Files (x86)\Microsoft Visual Studio 9.0

   Are these correct [y]/n?

4. Then press "y" and enter. This sets up the compiler to be used.
5. Next type "*mex filename.c*", this will generate the mexfile that can be used in other programs in MATLAB.

APPENDIX B

PCA ALGORITHM (MATLAB CODE)

```
tic;
clc;
clear;
%% enter the data%%
n=4801;  %%%____ Number of obseravations of each variable___%%%
m=41;  %%%%________ Number of variables_________%%%
t=0:1:4800;
%% Enter the data to train the model here. Each variable in each row %%
% here we are reading the data from the text file
fid = fopen('Normal_48.txt');
X = fscanf(fid, '%g', [m n]);
fclose(fid);

for i=1:m
   for j=1:n

    P(i,j) = X(i,j)- mean(X(i,:));
   end
end
for i=1:m
   for j=1:n
      P_std(i,j) = P(i,j)/std(P(i,:));
   end
end
%now the data has zero mean and unit variance

%% to find the covariance matrix C and calculating the loading vectors using
%Singular Value Decomposition (SVD)
% C is a 41x41 square matrix

for i=1:m
    for j = 1:m
        C(i,j) = sum(P_std(i,:) .*P_std(j,:)) / (n-1);
    end
end
 [U,S,V] = svd(C);

%% --------to find the the number of principle components ---
%%The number of principal components are found to be equal to 11 using
%%parallel analysis method(raw.m program)
a = 11;
L = V(1:m,1:a);
%% transforming data back to original coordinates
T=L'*X;
X_final = L*T;%% back transformation
%% Reading data with noise
% here we are reading the data from the text file
f1id = fopen('IDV1_48.txt');
```

```
X_noise = fscanf(f1id, '%g', [m n]);
fclose(fid);

T_noise=L'*X_noise;
X_final_noise = L*T_noise;%% back transformation

figure(1)
subplot(2,1,1)
plot(t/100,X(2,:))
hold on
plot(t/100,X_noise(2,:),'r')
hold off
subplot(2,1,2)
plot(t/100,X_final(2,:))
hold on
plot(t/100,X_final_noise(2,:),'r')
hold off

figure(2)
subplot(2,1,1)
plot(t/100,X(19,:))
hold on
plot(t/100,X_noise(19,:),'r')
hold off
subplot(2,1,2)
plot(t/100,X_final(19,:))
hold on
plot(t/100,X_final_noise(19,:),'r')
hold off

figure(3)
subplot(2,1,1)
plot(t/100,X(26,:))
hold on
plot(t/100,X_noise(26,:),'r')
hold off
subplot(2,1,2)
plot(t/100,X_final(26,:))
hold on
plot(t/100,X_final_noise(26,:),'r')
hold off

figure(4)
subplot(2,1,1)
plot(t/100,X(32,:))
hold on
plot(t/100,X_noise(32,:),'r')
hold off
subplot(2,1,2)
plot(t/100,X_final(32,:))
hold on
plot(t/100,X_final_noise(32,:),'r')
hold off
```

APPENDIX C

PARALLEL ANALYSIS

Numerous data studies use PCA for dimensionality reduction. The crucial problem is determining the number of principal components to be considered for forming the score matrix (T). An incorrect choice of method for finding the number of principal components can lead to under extraction or over extraction from the data. Among the several methods that were mentioned in Section 3.2, parallel analysis was proven accurate consistently in determining the exact cutoff of the number of principal components. In parallel analysis, many data sets are generated by rotating the original data from the plant. Then the eigenvalues of the raw data from the plant are compared with those from a normal pseudorandom deviates of same dimensionality to the original data set (same number of samples and variables). The eigenvalues of the original data greater than the corresponding eigenvalues from the random data sets are retained.

The program (M-file) that can be used to get the number of principal components can be downloaded from the following website.

https://people.ok.ubc.ca/brioconn/nfactors/nfactors.html (last accessed on 10/16/11).

VITA

Vamshi Krishna Kandula is a graduate student in the Department of Electrical Engineering at Louisiana State University. He received his bachelor of technology degree in electrical and electronics engineering from Vellore Institute of Technology, Tamilnadu, India, in May 2009. He will be graduating with the degree of Master of Science in electrical engineering in December 2011. He will be joining University of Florida to pursue further higher studies as a PhD in spring 2012.